Méthodes Empiriques et Langages de Script
TP 2

Gabriele A. Musillo
musillo4@etu.unige.ch

November 14, 2005

1 Exercice 1: le commande *tr*

Au moyen de la commande `tr`, résolvez les problèmes suivants:

- transformer les caractères majuscules d'un texte en caractères minuscules
- substituer à toute occurence de voyelle le caractère `V`
- substituer à toute occurence de consonne le caractère `C`
- supprimer les signes de ponctuation d'un texte
- afficher toutes les séquences de consonnes qui apparaissent dans un texte

2 Exercice 2: la tokenisation

Tokeniser un texte, c'est retourner les unités ou *tokens* qui le constitue. Les unités qui sont pertinentes à cet exercice sont les mots. Il n'est pas trivial de segmenter un texte en mots. Afin de simplifier la tokenisation, on stipulera qu'un mot est une séquence de caractères alphabétiques entre deux espaces. Téléchargez (au moyen de la commande `wget http://www.latl.unige.ch/mels/austen.1024.txt`, par exemple) le fichier `austen.1024.txt` et résolvez les problèmes suivants au moyen des commandes `tr`, `sort`, `uniq` et `head`:

- afficher un mot par ligne
- trier les tokens par ordre alphabétique et les dénombrer
- lister les 32 mots les plus fréquents ainsi que leur fréquence

3 Exercice 2: les n-grammes

- Un n-gramme est une séquence de n mots consécutifs. Donnez la séquence de commandes qui permet d'afficher tous les bi-grammes du fichier `austen.1024.txt` ainsi que leur fréquence (indications: les commandes `tail +2` et `paste` suffisent pour former tous les bi-grammes).

- Téléchargez le fichier `wsj_ptb.pos.txt`. Chacune des lignes ce fichier contient un mot suivi de son tag qui indique sa catégorie syntaxique. Donnez la séquence de commandes qui permet d'afficher toutes les séquences de trois tags consécutifs. Spécifiez également les commandes qui permettent d'afficher les séquences de trois tags qui apparaissent au moins 10 fois dans le fichier.